



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

09/682,315

08/18/2001

David Ge

2625

29390

7590

04/19/2007

DAVID GE

10218 125TH AVE. NE

KIRKLAND, WA 98033

EXAMINER

CHAVIS, JOHN Q

ART UNIT

PAPER NUMBER

2193

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
--	-----------	---------------

3 MONTHS

04/19/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No. 09/682,315	Applicant(s) GE, DAVID	
	Examiner John Chavis	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 17 January 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-3, 6 and 7 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-3, 6 and 7 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 08 March 2006 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 1-3 and 6-7 are rejected under 35 U.S.C. 112, first paragraph, as based on a disclosure, which is not enabling. The feature of "creating an application program in a codeless manner without using a compiler" is critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976). The applicant's response has been fully considered; however, the response address many features which are not present in the claims and the newly added feature of creating an application program in a codeless manner without using a compiler does not appear to be clearly supported by the specifications and therefore is considered new matter. The applicant mentions codeless in the specifications; however, it is not clear what makes the application codeless. This argument is considered to be supported by step A of the claim which states "defining and supporting a set of pre-developed object classes...". It is not clear what feature in the specifications makes the defining of pre-developed object classes codeless.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-3 and 6-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel in view of Budd and further in view of the applicant's choice of sorting a list of items to display. The applicant's response has been fully considered; however, the response address many features which are not present in the claims and the newly added feature of creating an application program in a codeless manner without using a compiler does not appear to be clearly supported by the specifications and therefore is considered new matter. The applicant mentions codeless in the specifications; however, it is not clear what makes the application codeless. This argument is considered to be supported by step A of the claim, which states "defining and supporting a set of pre-developed object classes..." It is not clear what feature in the specifications makes the defining of pre-developed object classes codeless. Therefore, the previous rejection is still considered appropriate. However, although the codeless feature is not considered to be supported by the applicant's specifications, the feature will be addressed below in view of the newly cited references. **The previous action is hereby repeated with bold references to address the newly presented arguments.**

US-CL-CURRENT: 717/113

CLAIMS:

1. A method for creating an application program...the method implemented on a

Patel/Budd/choice of sorting list

Each of the references provide for creating an application. Patel

computer system having persistent storage,

indicates it via pages 265-268.

The persistent storage is illustrated via the stored functions and properties available for modification in figure 15.1.

The applicant claims that his system functions in a codeless manner without using a compiler; however, it is not clear where this feature is taught in the specifications. The applicant merely mentions "codeless", as indicated below and it is not clear where or if a reference is made to a "compiler".

The applicant's major focus appears to be in "visual programming". This is also the focus of Patel, as indicated on page 264. On page 264 (the first paragraph), Patel indicates that a builder tool is a software environment in which you can assemble Java Beans-for example, a visual application builder...

What you find in these software environments is a palette of beans to choose from that can be "visually manipulated on-screen to form some now functional application (codeless programming); note also that Patel makes no reference to "compiling".

He further specifies on page 265, third paragraph, that the BeanBox is a free-

form container that allows beans to be dropped in and manipulated. The property sheet (list) shows all of the editable properties for the currently selected Bean. Furthermore, see the example referencing Juggling Beans (beginning on page 265) and note again that no reference is made to “ compiling” . That is, Patel enables beans to be connected and thereafter events to be fired. The need for compiling is considered specifically unnecessary when the code is already developed, see page 269 and the first paragraph of page 275.

a display screen and one or more input devices,

See again fig. 15.1 and the feature that enables the selections provides for input.

the input devices controllable by a user to create visual representations on the display screen,

See the last paragraph of page 265 and note that Patels GUI provides for visual representation. Also, the paragraph inherently provides for an input device to enable functions to be started, stopped and changed.

the method comprising...

A. defining and supporting a set of

The Toolbox in the last paragraph of page 266 provides for the pre-

pre-developed object classes, the said pre-developed object classes are all derived from one generic class which supports a property-method-event model;

defining an action class and an action list class; the action class has, as its members, action performer, action method, and action data;

developed object classes. The one generic class in Java is the Object class, see page 126 of the attached Linden reference, which is merely cited to provide a definition of objects. Note in the last paragraph that Linden indicates that " Object" is a superclass of all other classes in the system (generic).

Patel does not specifically indicate action classes and action lists specifically. However, it is not clear what the defining of specific classes (code) and list has to do with (codeless) programming. That is , what makes the system codeless? The applicant' s specification indicates that the user builds the action list (see sect. 0066), which does not appear to be codeless. The applicant mention " codeless programming " in sect. 0117 of the specification; but, does not show or teach how it is performed.

The applicant further indicates that his system comprises codeless programming without compiling; however, it is not clear which language enables the development (defining) of a

specific class without requiring compiling. The applicant further indicates that his action class (code) and action list class (again code) are key components of his codeless programming system. However, as previously indicated the two appear to conflict.

However, the features are taught by Budd to enable interfaces and user interactions, see page 91 (the actionPerformed and ActionListener methods, the FirebuttonListener class on page 92 and the data that supports each). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the feature for the same reasons in Patels system to enable dynamic user interactions, see the last two paragraphs of page 89. The action list class is considered taught via Budds menu items on pages 227-231 implemented like containers. The menu items are provided to simplify user selections and therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention

to utilize the feature in Patels system for the same reason.

the action list class contains a sorted list of action class instances;

Budds list or menu items are not listed as sorted; however, it would have been obvious to a person of ordinary skill in the art at the time of the invention to provide a sorted list to Patels system to enable users to visually locate selections faster.

the action performer is one of the pre-developed object class;

ActionListener functions as the action performer class; since, it must implement the actionPerformed method, see Budds section 6.2.2. The class is pre developed with it inherent constraints as indicated above and in section 6.2.2.

the action method is one of the methods supported by the action performer;

See again section 6.2.2, which provides for the actionPerformed method (action method).

the action data are the parameters needed by the action method;

See the ActionEvent data required by the actionPerformed method in sect. 6.2.2.

B. generating and graphically displaying, in response to input from the user,

See the instances displayed on page Budds 212, as an example and

instances of the object classes from the said pre-developed object classes;

specifically see Patels fig. 15.1.

C. setting, in response to input from the user, each property of each instance of the object classes created in step B;

See Patels figs. 15.1-15.4 for the property setting feature.

2. The method of claim 1 further comprising the step of: D. creating, in response to input from the user, instances of the action list class which contains a sorted list of instances of the action class;

See the rejection of claim 1.

wherein step D comprises

See Patels figs. 15.1-15.4.

D1. Creating, in response to input from the user, each action class instance of each action list class instance; and

Wherein step D1 comprises:

“ “ “ “

D1a. Selecting an instance of object class from the instances of object classes created in step B; the said selected instance of object class is used as the action performer member for the action class instance;

D1b. selecting a method from the methods supported by the instance of

“ “ “ “

object class selected in step D1a;

the selected method is used as the
action method for the action class
instance;

D1c. according to the method selected
in step D1b, it is known the number
and types of the parameters needed for
the said method; if one or more
parameters are needed for the said
method, then one or more dialog-
boxes are provided for the user to
specify the appropriate parameters for
the method;

3. The method of claim 1 further
comprising the step of E. linking, in
response to input from the user, action
list instances created in step D to
events of the instances of the object
classes to form an event-action-list
mapping...;

Wherein step E comprises:

E1. Selecting, in response to input
from the user, an instance of object
class from the existing instances of

“ “ “ “

The feature is not specifically
indicated by Patel; however, see Budds
fig. 13.5. Budd utilizes the feature to
enable the user to know what input
is required or supported. Therefore,
it would have been obvious to a
person having ordinary skill in the art
at the time of the invention to utilize the
feature in Patels system for the same
reasons.

See Patels fig. 15.7.

See Patels fig. 15.15.

object classes;

E2. Selecting, in response to input from the user, an event from the events supported by the object class instance selected in step E1;

" " " "

E3. Selecting, in response to input from the user, an action list class instance from the action list class instances created in step 2;

See fig. 15.9 in view of fig. 15.6.

E4. Building the mapping relationship between the action-list selected in step E3 and the event selected in step E2;

See again figs. 15.6 and 15.7.

4. The method of claim 1 further comprising the step of: F. Selecting, in response to input from the user, a set of object class instances to be specified as the "initially active object class instances" usually the object class instances presented on the first application screen presentation is such a set of the "initially active object class instances"

See fig. 15.6 and note that either start, stop or neither has to initially be set to active.

5. The method of claim 1 further

This claim is so unclear that the

comprising the step G. saving to the computer persistent storage the object class instances created in steps A, B and C, the action list class instances created in step D, the mapping relationship built in step E between the events of object class instances and the action lists, indication of which object instances are the "initially active object class instances" as specified in step F;

features not supported are not entitled patentable weight.

Therefore, the claim is rejected as claim 1.

6. The method of claim 1 further comprising the step of: H. an execution environment; Wherein step H comprises: H1. Reading back from the computer persistent storage the object class instances created in steps A, B and C, the action lists created in step D, the mapping relationship built in step E between the events of the object class instances and the action list, indication of which object instances are the "initially active object class instances" as specified in step F; H2. Creating and displaying the said "initially active object class instances" H3. Responding to each event fired by each object class instance; Wherein step H3 comprises: H3a. Checking if there is a mapping

See the rejection of claim 5 above.

relationship between an action list class instance and the said event; H3b. If the said mapping relationship exists, sequentially performing each action in the said action list mapped to the said event; H3c. Each action in the said action list is performed by the following steps: H3c1. Locating the object class instance which is assigned as the action performer for the action; H3c2. Signaling to the said action performer which action method is specified for the action; H3c3. If there are method data specified for the said method of the said located object class instance, the method data are passed to the said object class instance as well; H3C4. The said located object class instance carries out the said action method.

7. The method of claim 1 further comprising the step of: I. A context-data buffer which saves event parameter data such as mouse position in mouse movement events; every time an event is fired, before an action list is executed as an event handler, the said context-data buffer is filled with the said event parameter

“ “ “ “

data; J. The said context-data buffer is available for the user to pick as the method data in step D1c;

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

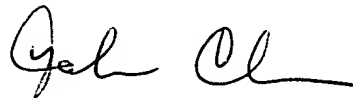
6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to John Chavis whose telephone number is (571) 272-3720. The examiner can normally be reached on M-F, 9:00am-5:30pm, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2193

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JC

A handwritten signature in black ink, appearing to read "John Chavis", with a stylized flourish at the end.

John Chavis

Primary Examiner AU-2193